

# Biologically-Inspired Condition Monitoring of Civil Engineering Structures

Dr. Kay Smarsly, *Senior Member, IACSIT*

**Abstract**—Long-term deterioration of civil engineering structures is a serious problem of crucial societal and economical relevance. In these years, civil engineering structures such as bridges or wind turbines are reaching or exceeding their original anticipated lifespan. Consequently, there is an urgent need for applying modern structural health monitoring systems to ensure serviceability and structural safety through efficient as well as cost-effective monitoring. In this contribution, a novel generation of biologically-inspired structural health monitoring systems is introduced, synergistically coupling sophisticated engineering concepts and technologies stemming from artificial intelligence research. To exemplify the proposed concepts in more detail, the design and the implementation of a biologically-inspired and self-managing (i.e. self-configuring, self-optimizing, self-protecting and self-healing) structural health monitoring system prototype is presented. Practicability, reliability and performance of the prototype are corroborated by means of structural and environmental data taken from different engineering structures in tandem with intensive laboratory tests.

**Index Terms**—Biologically-Inspired Structural Health Monitoring, Computational and Artificial Intelligence, Multi-Agent Technology, Sensor Networks, Emergent Computing.

## I. INTRODUCTION

Autonomic Computing, Artificial Life Research, Organic Computing and Biologically-Inspired Computing are active areas of research that represent novel scientific disciplines. Their overall goal is to mimic and then to formalize processes observed in living systems by translating them to computing methods for solving highly sophisticated as well as complex problems appearing in different sciences. To give an example, Collective Intelligence, commonly referred to as Swarm Intelligence, can be observed in manifold natural systems such as ant colonies, schools of fishes or flocks of birds representing emergent phenomena. In general, a population of interacting entities follows simple rules in absence of a centralized control structure leading to the emergence of “intelligent” global behavior, which is unknown to the individual entities (s. Fig. 1).

Since engineering problems have become increasingly complex, these inherently intelligent structures and the emergent behavior of biological organisms, which are used as demonstration models (role models), can inspire engineering disciplines to create decentralized and self-organized information processing systems. While epistemologists have examined the emergent behavior in

complex systems for more than a century, in engineering disciplines the investigation and the application of emergent systems, created by the merging of Biology and Computer Science into practicable, robust and cost-efficient systems, is still in its infancy [1, 2]. In order to achieve an alive and fault-tolerant behavior of engineering systems, various approaches can be applied to gain biologically-inspired computer solutions such as

- 1) Artificial Neural Networks,
- 2) Evolutionary Algorithms,
- 3) DNA Computation,
- 4) Cellular Automata,
- 5) Amorphous Computing,
- 6) Computer Immune Systems, etc.

Relying on well-established biological and social paradigms, large quantities of small processing units, in particular sensors and actuators can be implemented in engineering systems that show capabilities of self-management. By that, complex system behavior at the macroscopic level can be mapped in a non-linear fashion derived from the heterogeneous interactions on the microscopic level.

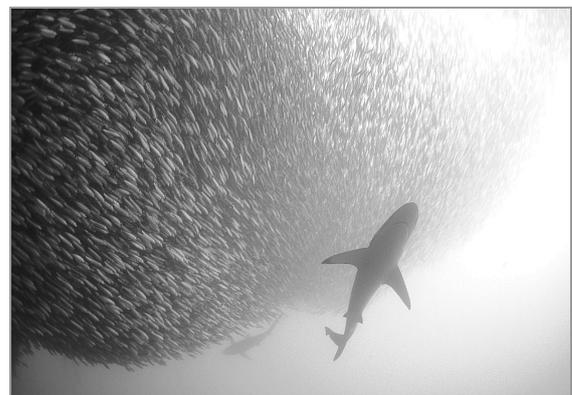


Figure 1. Swarm behavior of a school of sardines surrounded by copper sharks (source: [3])

With respect to biologically-inspired solutions, considerable progress has recently been made in many areas. To give some examples, Cellular Automata are used for solving mathematical, physical or biological problems, self-learning Artificial Neural Networks are employed for identifying individuals by means of facial recognition systems that imitate human brain functions [2]. Furthermore, mechanisms observed in biological evolution processes such as reproduction, recombination, mutation and natural selection are effectively introduced for solving nonstandard optimization problems based on Evolutionary Algorithms

[4, 5]. In Structural Health Monitoring (SHM), however, biologically-inspired computing has not yet been considered coequally. In this paper, a new approach towards SHM is introduced after giving a brief introduction on a biologically-inspired SHM concept proposed in accordance to natural design principles. Thereupon, the design, the implementation and the validation of a biologically-inspired SHM system prototype is demonstrated. Providing autonomous structural controls and proactive support for the engineer in charge of monitoring, the SHM system prototype materializes first decisive biologically-inspired concepts as illuminated in the course of the paper.

## II. A BIOLOGICALLY-INSPIRED APPROACH TOWARDS STRUCTURAL HEALTH MONITORING

One of the main goals of Structural Health Monitoring is to determine the actual conditions of engineering structures, such as bridges, dams or wind turbines, to identify deteriorations and damages at the earliest possible stage. The urgent demand for capable SHM systems and the need for comprehensive safety assessment strategies are highlighted by recent collapses of public engineering structures. Fig. 2, for example, shows severe collapses of diverse types of engineering structures that occurred in India, Germany and Indonesia in the year 2009. Evidently, these incidents could have been avoided by installing commonplace SHM systems capable of permanently analyzing the actual conditions of the structures. The current state of practice, however, relies on economically inefficient and more or less subjective maintenance routines, periodically performed in large intervals.



Figure 2. Examples of severe collapses of different types of engineering structures in 2009: (a) chimney in Korba, India, (b) historical archive in Cologne, Germany, (c) dam in Jakarta, Indonesia, (d) metro bridge in New Delhi, India (sources [6-9])

Similar to living systems that permanently sense and analyze stimuli coming from their environment, SHM

systems continuously sense and analyze structural as well as environmental quantities measured in the observed structure (e.g. displacements, vibrations or temperatures). To this end, the gauges are collected and pre-processed in real-time through the sensor systems installed in the structure, transmitted to decentralized computer systems for further analyses and, finally, stored in databases accessible to the human experts charged with monitoring. As a result, a drastically advanced lifespan assessment as well as an improved safety can be achieved, in parallel to significantly reduced maintenance costs of aging structures.

In the past years, considerable progress could be achieved in improving “conventional” monitoring strategies by advanced SHM approaches and practical SHM systems. For instance, wireless solutions are coming up, new data communication concepts are proposed [10-16], and related problems such as energy consumption of wireless devices and consistent interaction mechanisms are increasingly being resolved [17-20]. Beyond that, the progress recently made in related disciplines might in future considerably advance solutions of specific SHM problems. In this context, data analysis techniques [21], machine learning algorithms [22], data mining [23] and regression techniques [24] have recently been investigated and optimized. Furthermore, substantial advancements have been made in further disciplines that are closely connected to SHM research, such as artificial intelligence systems [25], high-performance computing [26], biologically-motivated machine vision [27] and handling uncertainty in complex engineering problems [28]. In summary, SHM research represents a highly attractive research field associated with substantial benefits to the society.

The fundamentals of biologically-inspired and self-managing SHM systems can rest on multi-agent technology where software agents are seen as cooperative “problem solving entities” that are situated on decentralized computers within a computer network: According to the organizational structure of a multi-agent system similar to biological systems, each software agent is responsible for solving a specific monitoring task, such as continuously analyzing structural data, automatically creating safety reports or permanently conducting system self-diagnoses. For solving these monitoring tasks appropriately, different biologically-inspired concepts can be utilized by the software agents. For example, self-diagnosis of the SHM system can be realized through Artificial Neural Networks capable of learning new problem-solving. For that reason, occurring events or changes of system states within a safety-relevant SHM system (e.g. sensor malfunctions, unavailable network resources, etc.) are permanently detected and, in the case of occurrence, resolved in real-time avoiding a total system collapse as may happen in conventional SHM systems. Further use cases of biologically-inspired concepts are the utilization of Fuzzy Logic embedded into the software agents for handling of incomplete or uncertain information, or the application of Evolutionary Algorithms for determining the best (optimal) sensor locations by the multi-agent system.

Following [29], a multi-agent system *MAS* describes a system that consists of

- 1) an environment  $E$  that possesses a volume  $V$ ,
- 2) a set of objects  $Obj$  situated in  $E$ ,
- 3) an assembly of agents  $A$  representing active objects ( $A \subset Obj$ ),
- 4) an assembly of relations  $R$  linking  $Obj$ ,
- 5) an assembly of operations, methods and operators  $Op$  allowing  $A$  to perceive, produce, transform, manipulate and delete objects through sensors and actuators.

Consequently, the monitoring environment,  $E_{SHM} = \{e_0, e_1, \dots, e_n, \dots\}$  and the environmental states  $e_i$  can be changed by the agents executing agent actions  $Ac = \{\alpha_0, \alpha_1, \dots, \alpha_m, \dots\}$  either reactively or proactively. The states  $e_i$  are then dynamically modified and adapted to continuously changing environmental conditions. With respect to realizing fully autonomous SHM systems, the sets of operators  $Op$  and methods  $M$  can be extended by additional sets of operators  $Op_{self}$  and methods  $M_{self} = \Sigma M_{self,i}(A_i)$  that allow self-configuration, self-optimization, self-healing and self-protection of agent-based SHM systems as proposed in [30], where

$$M_{self} \subset M; Op_{self} \subset Op. \quad (1)$$

As a result, complex system behavior can emerge from single agent behaviors in the absence of central control. This phenomenon is referred to as “emergent behavior”  $Em$  to be utilized as a basis for generating Collective Intelligence  $CI$ . Thus, according to [30], the corresponding function composition for evoking collective group behavior in agent-based SHM systems can be defined as

$$(CI \circ Em)(MAS) = CI(Em(MAS)). \quad (2)$$

### III. DESIGN AND IMPLEMENTATION OF A SHM SYSTEM PROTOTYPE

Nucleus of the agent-based SHM system is a hybrid multi-layer model  $HMLM$  that synergistically couples two different paradigms being materialized in two SHM subsystems (s. [31]). On the one hand, intelligent software agents  $A$  (software *entities*) situated on different computers form a distributed multi-agent system  $MAS_s$ . On the other hand, connected to the multi-agent system collaborating smart sensing units (*hardware entities*) based on microcontroller technology are to be embedded in the observed engineering structures constituting a multi-agent system based on hardware agents  $MAS_h$ .

As to that, the system development follows an explicit design methodology for agent-based SHM systems proposed by Smarsly [32, 33]. As a result, a consistent SHM system model is achieved which is composed of so-called “monitoring domains” as outlined in Fig. 3. Taking into account the application of software and hardware agents, two categories of monitoring domains can be distinguished: Agent-based domains  $DA_i$  and a microcontroller-based sensing (i.e. hardware) domain  $DM_s$ , whereby every entity involved is explicitly assigned to one specific monitoring domain. According to Fig. 3, the agent-based cooperation domain  $DA_c$  encapsulates cooperation agents that serve as proactive personal assistants to the human experts involved.

Mapping the real world network, one cooperation agent is assigned to one human actor as a “personal assistant” and is installed locally on the particular computer, either on a PC or on a mobile device. Secondly, the agent-based adaptation domain  $DA_a$  ensures a flexible transferability of the SHM system to different structure categories (e.g. tunnels, dams, towers, bridges). In this context, required meta-information according to different monitoring projects is provided within  $DA_a$ . The process domain  $DA_p$  comprises process agents  $A_{p,i}$  that are crucial for autonomous execution of the monitoring processes. At last, the wrapper agents  $A_{w,i}$  of the integration domain  $DA_w$  integrate non-agentic software as well as hardware into the agent-based SHM system such as FE software, database systems and sensing units.

Incorporated into the sensing domain  $DM_s$  and connected to the agent-based software system, the artificial society of interacting smart sensing units is applied where  $MAS_h = MAS_h(DM_s)$ . Fig. 4 illustrates one of the smart sensing units forming in total  $MAS_h$ . Based on the JControl technology [34, 35], the smart sensing units prototypically being deployed allow a Java-based implementation of object-oriented, robust and scalable hardware agents.

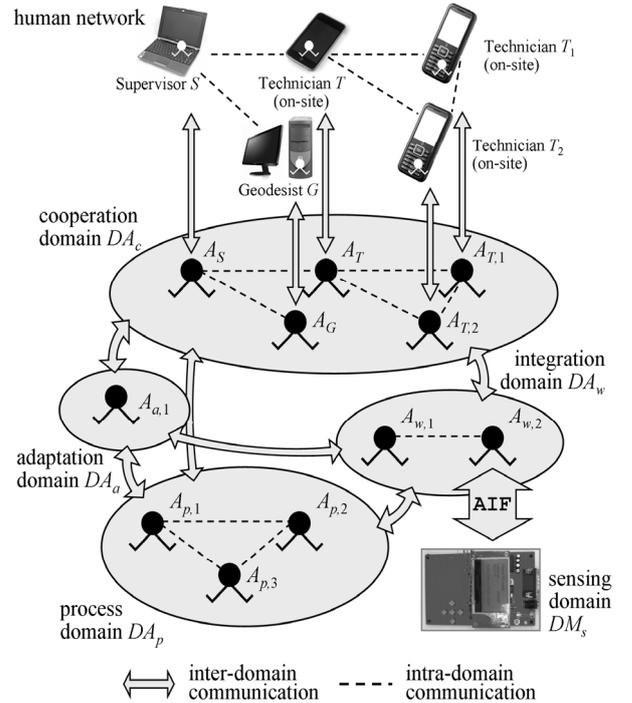


Figure 3. Decentralized architecture of the monitoring system

Their purpose is to acquire structural and environmental data autonomously, to preprocess the collected data through embedded algorithms and to communicate data as well as internally computed analysis results to the agent-based software system  $MAS_s$ . To this end, the so called “AMBOS Interchange Format” (AIF) is defined to compress and exchange monitoring data efficiently between the smart sensing units and the software agents (details can be found in [32]). Taking into account the dualism of the hardware and the software subsystem, a decentralized monitoring system is defined as a function of the hybrid multi-layer model

$$HMLM = \Sigma DA_i + DM_s. \quad (3)$$

Consequently, the achieved multi-agent system can be written as

$$MAS = MAS(HMLM) = \Sigma MAS_i. \quad (4)$$

Extending Eq. 2,  $Em$  is defined as a function of  $HMLM$ , i.e.

$$Em = Em(MAS(HMLM)) \quad (5)$$

and  $CI$ , introducing a composite monitoring operator  $C_{SHM}$  that maps the monitoring functions  $CI$  and  $Em$  associated with collective group behavior, can thus be written as

$$C_{SHM}CI = CI \circ Em. \quad (6)$$

Finally, to achieve collective group behavior, every entity, both software agent and smart sensing unit, fulfills one or more specific roles, provides comprehensive monitoring services, performs complex interactions and stands in certain relationships with several other beings. Therefore, various submodels to the hybrid multi-layer model are defined, such as roles model, services model, interactions model, acquaintance model, etc. (s. [32, 33]).

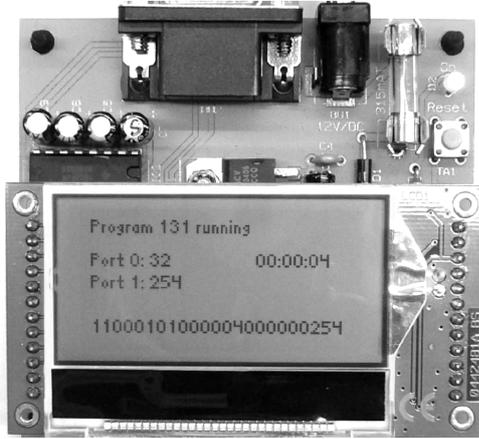


Figure 4. Java-based smart sensing unit (cf. [32, 34, 35])

#### A. Autonomous workflow control

Autonomous workflow control and system self-management are realized by a two-tiered approach that interlinks the concepts of central planning and distributed problem solving. On a coordination layer, agent  $A_S$ , the personal assistant to the human project supervisor  $S$  (s. Fig. 3), acts as a central planning unit and coordinates the complete monitoring workflow  $N$  by using Petri net technology. On a conversation layer, each emerging monitoring process is dynamically translated into congruent agent interactions through transformation rules. Thereby, each specialized agent decides autonomously how to solve the current process in a decentralized, self-contained fashion in terms of agent services, interactions and behaviors using well-defined interaction protocols and agent message exchange based on the Agent Communication Language (ACL) in compliance with the specifications of the Foundation for Intelligent Physical Agents (FIPA [36, 37]).

#### B. System self-management

In conventional real-time systems, exceptions caused by abnormal events, disturbances or failures that may occur during task execution are usually logged. Thereupon, the

whole task is aborted. At worst, a complete system failure is created. Typical examples known from SHM systems are

defect sensors, interrupted database connections or unavailable servers. Such exceptions frequently result in the loss of valuable data or inaccurate safety assessments. To solve these difficulties, novel computer systems try to predefine an individual strategy for handling each exception, e.g. using expert systems. However, such rule-based approaches require detailed knowledge of all possible exceptions in advance, which is impossible in dynamic large-scale systems. Consequently, a priori unknown exceptions could not be handled applying these strategies.

As opposed to those customary approaches, the self-management introduced in this paper is based on a fault-tolerant strategy that takes corrective steps if exceptions appear. Hence, its purpose is not to generally avoid every exception, but to react appropriately on exceptions, to compute alternative tasks and, finally, to detect and to correct the causes of the exceptions automatically. This strategy is implemented in a self-healing module based on Artificial Neural Networks as a computational part of the supervisor agent  $A_S$ . In detail, every regular monitoring task  $t'$ , regularly conducted automatically by the agents, is assigned to a control block  $k$  (s. Fig. 5). A control block represents a functional unit that summarizes coherent sets of monitoring tasks  $t' \in T$  and system states  $s' \in S'$  within a certain monitoring workflow  $N' = N'(S', T', F')$  being part of the global workflow  $N$  where  $N' \subset N$ .

Exceptions occurred are detected by the supervisor agent  $A_S$ , e.g. by means of simple timeout mechanisms or non-responding hardware or software entities. The affected monitoring task  $t'$  and the corresponding control block  $k$  where  $t'$  is assigned to, is identified by  $A_S$  in real-time. Exceptional monitoring tasks  $X_k$  are processed by  $A_S$  alternatively instead of the invalid monitoring task  $t'$  to keep the system running in a first step (s. Fig. 5, "step  $i$ "). Thereupon, the cause of the exception is calculated in order to sustainably fix the problem: A specialized, polymorphic agent  $A_P$  is replicated and incorporated into the agent-based system at runtime. For identifying the exception cause,  $A_P$  constructs a two-layered neural network (type pattern associator) that is capable of learning and allocating associations between input and output patterns reliably even if data is fuzzy or incomplete (s. Fig. 5, step "ii"). Input patterns are understood as observations, i.e. the effects of the exception that the agent perceives (e.g. the message "no data available"). Output patterns represent identified exceptions which are the causes calculated by the neural network, such as "database agent malfunction" or "network connection interrupted". Both input and output patterns are internally mapped by discrete numerical values for an efficient processing. For that, the pattern associator network is trained to respond with a certain output pattern (cause) when an input pattern (observation) is invoked. In this case, previously simulated cause-observation pairs, stored in a monitoring case base, are applied. The connection weights between the input and output layer, representing the knowledge (weight matrix  $\mathbf{W}$ ) of agent  $A_P$  (s. Fig. 5), are adjusted through the

training routines for self-optimization of the input/output behavior.

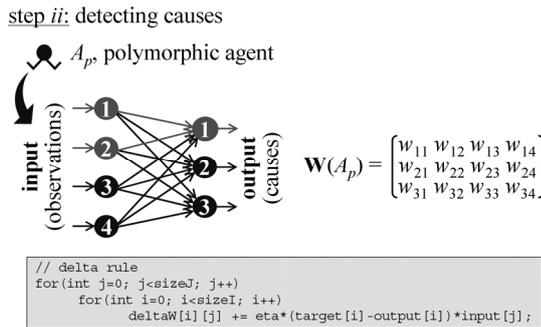
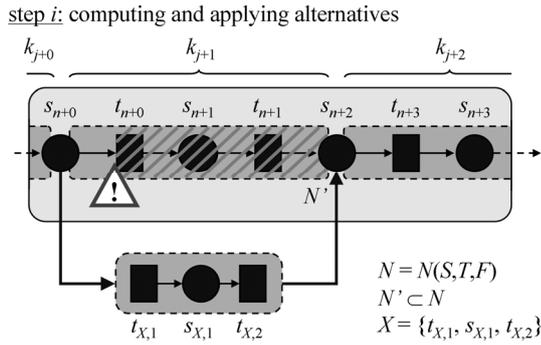


Figure 5. Principle of self-healing applied by the software agents

At last, the network is applied to the new situation. As a result, the exception cause is determined by the network output and, thereupon, autonomously corrected by  $A_p$  if possible, or in cooperation with the human actors and the other agents deploying  $M_{self}$  and  $O_{pself}$  (cf. Eq. 1).

#### IV. SYSTEM VALIDATION

Intensive validation tests of the implemented SHM system prototype are conducted on a laboratory level particularly focusing on the aforementioned self-management capabilities. Fig. 6 and Fig. 7 illustrate, as an example, the autonomous reaction of the system prototype on exceptional situations. To this end, the data acquisition agent  $A_{p,1}$ , responsible for continuously collecting and preprocessing structural data, is manually terminated in order to provoke a complete system collapse. In conventional systems, a workflow block would be created leading to a complete system failure and to the loss of monitoring data because a crucial step within the monitoring workflow – the data acquisition task – is no longer automatically performed. By contrast, the autonomous reaction of the system prototype behaves differently starting its self-managing mechanisms: In a first step, the specialized, polymorphic agent  $A_p$  is replicated by  $A_S$  and incorporated into the agent-based system at runtime (Fig. 6).

generated source of the polymorphic agent  $A_p$

individual, situation-specific agent attributes

new agent services provided by the replicated agent instance

updated agent population of the system

Figure 6. Agent replication for system self-diagnostics

For identifying the exception cause,  $A_p$  constructs the two-layered neural network. After having accomplished the training routines applying the monitoring case base providing discrete simulated *cause-observation* pairs (Fig. 7a), the cause for the generated exception, which can be manifold (“defect sensors”, “terminated agents”, “interrupted communication links”, etc.), is calculated (Fig. 7b). Then, the agent’s knowledge stored in the weights matrix  $\mathbf{W}$  is updated (Fig. 7c). Here, the induced exception is perceived through the *observations* “recent data unavailable” mapped in the simulation-based monitoring case base as “behavior 3” and “sensing units inaccessible” mapped as “behavior 8”. Representing the output of the neural network, the data acquisition agent  $A_{p,1}$ , which is formally defined by “case 4”, is with the highest degree of probability correctly identified as the *cause* for the exception. Finally, an individual recommendation for corrective actions is presented to the responsible engineer as shown in Fig. 7d by means of the corresponding GUI generated by  $A_p$ .

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0
4	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Figure 7a. Discrete numerical values constituting the monitoring case base (excerpt)

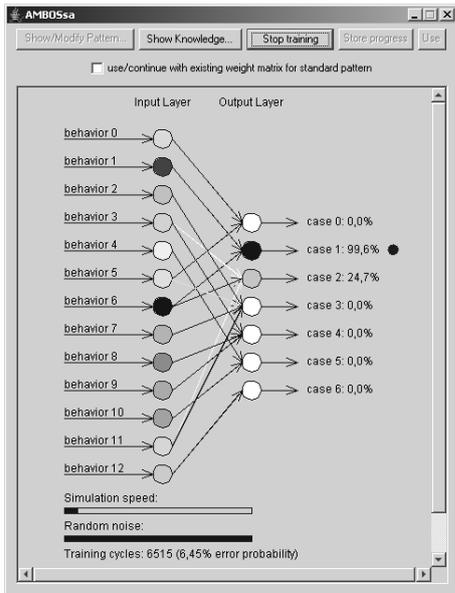


Figure 7b. Autonomous pattern recognition

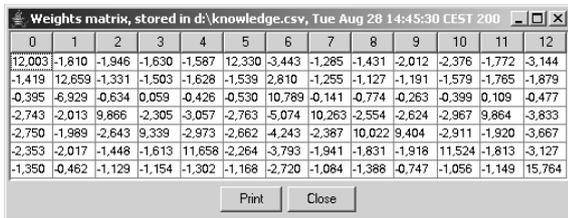


Figure 7c. Self-optimization of the agent's internal knowledge through modification of the weights matrix

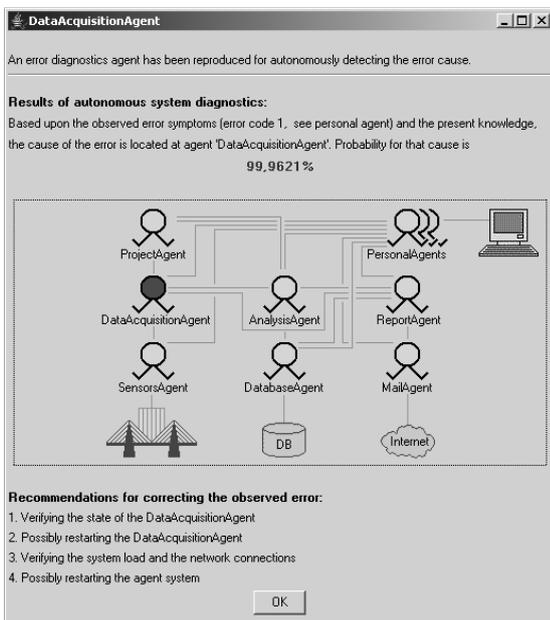


Figure 7d. Autonomous system diagnostics: Calculated cause presented to the engineer in charge in tandem with recommended actions for efficient problem solving

In addition to the self-managing capabilities, so called “conventional monitoring functions”, provided by the system prototype, are validated utilizing structural data taken from real engineering structures, particularly bridges, wind turbines and dams. At this, the Triebischseitental valley bridge, a box girder bridge with a length of 330 m located near Nossen (Germany) that has been erected in 1999, shown in Fig. 8 serves as a reference structure. To give some

examples, further comprehensive system tests are conducted with respect to essential functions (for details s. [32]), including e.g.

- 1) manual interaction mechanisms for communicating with the smart sensing units
- 2) continuous structural diagnoses and prognoses based on regression analysis, data mining techniques and probabilistic methods
- 3) permanent documentation of monitoring processes along with automated generating and mailing of safety reports
- 4) reliable remote access to software components such as integrated database systems and data analysis modules
- 5) proactive alerting in case of detected structural anomalies

The results indicate that notable benefits of biologically-inspired, fault-tolerant systems can be achieved in the field of SHM. In particular, a reliable monitoring can be provided leading to an increased structural safety through specialized cooperating software as well as hardware entities. In conclusion, employing these decentralized hybrid, i.e. hardware-software, SHM systems is estimated to result in a powerful concept for modern structural health monitoring.



Figure 8. Reference structure: Triebischseitental valley bridge

## V. CONCLUSIONS AND OUTLOOK

In this paper, the design, the implementation and the evaluation of a biologically-inspired SHM system prototype has been presented. In particular, system self-management, the intrinsic property that distinguishes autonomous systems from automated systems, has been elucidated. It has been demonstrated that multi-agent technology is an appropriate basis for creating biologically-inspired and self-managing SHM systems and, furthermore, for generating inherent autonomous capabilities such as self-healing, self-configuration, self-optimization and self-protection.

Future research will even improve the outcome significantly. Transferring the new approach into practice will obviously prevent potential multifaceted system failures because early minor deteriorations can be detected in time such that proper maintenance activities can be introduced. In a next step, the existing SHM concept is to be extended by incorporating emergent phenomena of a higher level. Most notably, coupling autonomous software systems with

wireless sensor networks enables intelligent, emergent systems for large-scale infrastructure monitoring. In this context, the implementation of self-organization, like that used in the nature, e.g. in ant colonies, in schools of fishes or in flocks of birds, is of interest. Commonly referred to as “Swarm Intelligence” or “Collective Intelligence”, this category of emergent system behavior might be helpful to discover more complex patterns within engineering structures – particularly in cases when conventional damage detection approaches are prone to be not sufficiently reliable. Here, the artificial society of software agents forming the agent system is to be supported by autonomous hardware agents acting as intelligent wireless sensor nodes distributed over the respective structure. By upgrading the present internal agent architecture by means of advanced emergence-oriented communication modules, the drastic increase of the collaboration efforts accompanied by an arising number of interactions between individual agents can be mastered efficiently. This improvement is due to the collective intelligence allowing for new levels of robustness, scalability and flexibility in complex SHM systems to be considered in the deserved detail within further research that is currently being conducted.

#### REFERENCES

- [1] Müller-Schloer, C. and Sick, B., 2006. Emergence in Organic Computing Systems: Discussion of a Controversial Concept. In: L. T. Yang, et al. (eds). *Autonomic and Trusted Computing*. Heidelberg, Germany: Springer Verlag.
- [2] Müller-Schloer, C., 2004. *Organic Computing*. Informatik Spektrum, 27(4), pp. 332-336.
- [3] Doubilet, D., 2007. Copper sharks and a school of sardines off South Africa. National Geographic Society, Hamburg, Germany.
- [4] Smarsly, K., Lehner, K. and Hartmann, D., 2007. Structural Health Monitoring based on Artificial Intelligence Techniques. In: L. Soibelman and B. Akinci (eds.). *Computing in Civil Engineering*. Reston, VA, USA, pp. 111-118.
- [5] Smarsly, K. and Hartmann, D., 2007. Artificial Intelligence in Structural Health Monitoring. In *The Third International Conference on Structural Engineering, Mechanics and Computation (SEMC) 2007*, Cape Town, South Africa.
- [6] The Sandesh Limited, 2009. Chimney collapse in Madhya Pradesh. The Sandesh Limited, Sandesh Bhavan, Lad Soc Road, Bodakdev, Ahmedabad – 38005, Gujarat, India. [Online] Available: <http://www.sandesh.com> [Accessed January 04, 2010].
- [7] DPA, 2009. Photo archives. Deutsche Presse-Agentur GmbH, Mittelweg 38, 20148 Hamburg, Germany. [Online] Available: <http://www.dpa.de> [Accessed January 4, 2010].
- [8] Xinhua News Agency, 2009. Indonesia's dam burst kills 77. Xinhua News Agency, 20F, Dacheng Plaza, 127 Xuanwumen St. (W), Beijing, China. [Online] Available: [http://news.xinhuanet.com/english/2009-03/28/content\\_11088274.htm](http://news.xinhuanet.com/english/2009-03/28/content_11088274.htm) [Accessed January 04, 2010].
- [9] Sulekha.com, 2009. Delhi Metro Bridge Collapse. Sulekha.com, New Media Pvt. Ltd., No. 484 & 485, Pantheon Plaza, 4th Floor, Pantheon Road, Egmore, Chennai - 600 008, India. [Online] Available: <http://newshopper.sulekha.com> [Accessed January 04, 2010].
- [10] Manjunatha, T. C. and Bandyopadhyay, B., 2009. Vibration control of Timoshenko smart structures using multivariate output feedback based discrete sliding mode control for SISO systems. *Journal of Sound and Vibration*, 326(1), pp. 50-74.
- [11] Kavitha, C. and Viswanatha, K. V., 2009. A hybrid Reliable Routing Technique (HRR) for wireless sensor networks. *International Journal of Computer Science and Network Security*, 9(3), pp. 1738-7906.
- [12] Lynch, J. P., Law, K. H., Kiremidjian, A. S., Carryer, E., Kenny, T. W., Partridge, A. and Sundararajan, A., 2002. Validation of a wireless modular monitoring system for structures. In *SPIE's 9th Annual International Symposium on Smart Structures and Materials*, San Diego, CA, USA.
- [13] Wang, Y., Sundararajan, A., Law, K. H., Kiremidjian, A. S. and Lynch, J. P., 2004. Wireless Monitoring for Structural Health Monitoring of Civil Structures. In *Proceedings of International Workshop on Integrated Life-Cycle Management of Infrastructures*, Hong Kong, China.
- [14] Kottapalli, V. A., Kiremidjian, A. S., Lynch, J. P., Carryer, E., Kenny, T. W., Law, K. H. and Leia, Y., 2003. Two-tiered wireless sensor network architecture for structural health monitoring. In *SPIE's 10th Annual International Symposium on Smart Structures and Materials*, San Diego, CA, USA.
- [15] Kurata, N., Spencer Jr., B. F., Ruiz-Sandoval, M., Miyamoto, Y. and Sako, Y., 2003. A Study on Building Risk Monitoring Using Wireless Sensor Network MICA-Mote. In *First International Conference on Structural Health Monitoring and Intelligent Infrastructure*, Tokyo, Japan.
- [16] Kijewski-Correa, T., Haenggi, M. and Antsaklis, P., 2005. Multi-scale wireless sensor networks for structural health monitoring. In *2nd International Conference on Structural Health Monitoring and Intelligent Infrastructure*, Shenzhen, China.
- [17] Rathore, A. K., Bhat, A. K. S. and Oruganti, R., 2008. A Comparison of Soft Switched DC-DC Converters for Fuel Cell to Utility Interface Application. *IEEJ Transactions on Industry Applications*, 128(4), pp. 450-458.
- [18] Law, K. H., Lynch, J. P. and Wang, Y., 2004. Wireless Structural Sensors using Reliable Communication Protocols for Data Acquisition and Interrogation. In *Proceedings of 42nd Allerton Conference on Communication, Control and Computing*, Allerton, IL, USA.
- [19] Wang, Y., Law, K. H. and Lynch, J. P., 2005. A Wireless Structural Health Monitoring System with Multithreaded Sensing Devices: Design and Validation. In *Structure & Infrastructure Engineering: Maintenance, Management, Life-Cycle Design & Performance*, Florence, KY, USA.
- [20] Lakeman, C. D. E., Fleig, P. F., DeGreeff, J. L. and Trainor, J. T., 2006. Hybrid power for wireless sensors. In *Proceedings of SPIE 6231, Unattended Ground, Sea and Air Sensor Technologies and Applications VIII*, Bellingham, WA, USA.
- [21] Shobha, G. and Shettar, R., 2009. Document Classifier Based on Statistical and Unsupervised Machine Learning Techniques. In *Proceedings of International Conference on Computational and Mathematical Sciences (ICCMS 2009)*, Paris, France.
- [22] Kaur, A., Singh, Y. and Malhotra, R., 2009. Comparative analysis of regression and machine learning methods for predicting fault proneness models. In *IJCAT 35(2/3/4)*, pp. 183-193.
- [23] Dehmer, M., 2005. Data Mining-Konzepte und graphentheoretische Methoden zur Analyse web-basierter Daten. In *Journal of Computational Linguistics and Language Technology (2005)*, pp. 113-141.
- [24] Mansour, N. and El-Fakih, K., 1997. Natural Optimization Algorithms for Optimal Regression Testing. In *Proceedings of the 21st International Computer Software and Applications Conference*. Washington, DC, USA.
- [25] Capanni, N. F., 2006. The functionality of spatial and time domain artificial neural models. Thesis. Aberdeen: The Robert Gordon University, School of Computing, Aberdeen, Scotland, UK.
- [26] Yang, L. T., Minyi, G., 2005. *High-Performance Computing*. Wiley Series on Parallel and Distributed Computing. Hoboken, NJ, USA: Wiley.
- [27] Min-Allah, N., 2009. Seminar Series on Biologically Motivated Machine Vision. COMSATS Institute of Information Technology, Islamabad, Pakistan.
- [28] Ebrahim, A. and Murphy, G. V., 2007. Adaptive Backstepping Control of a Speed-Sensorless Induction Motor Under Time-Varying Load Torque and Rotor Resistance Uncertainty. In *IEEE proceedings of 39th Southeastern Symposium on System Theory*, Mercer University, Macon, GA, USA.
- [29] Ferber, J., 1999. *Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence*. London: Addison Wesley.
- [30] Smarsly, K., 2010. Advanced Structural Health Monitoring based on Collective Intelligence. Research project funded by the German Research Foundation (DFG) through the research grant SM 281/1-1. Stanford University, Stanford, CA, USA.
- [31] Smarsly, K., 2010. An Autonomous Computing Approach towards Monitoring of Civil Engineering Structures. *Asian Journal of Civil Engineering*, 11(2), pp. 149-163.
- [32] Smarsly, K., 2008. *Autonome Überwachung sicherheitsrelevanter Ingenieurbauwerke*. 1st ed. Aachen, Germany: Shaker Verlag.
- [33] Smarsly, K. and Hartmann, D., 2010. Agent-Oriented Development of Hybrid Wind Turbine Monitoring Systems. In *The International Conference on Computing in Civil and Building Engineering jointly*

with The XVII Workshop on Intelligent Computing in Engineering, Nottingham, UK.

- [34] Böhme, H. and Telkamp, G., 2001. JControl – Einfache Implementierung und Evaluierung von eingebetteten Anwendungen mit JAVA. In E.I.S.-Workshop - Entwurf Integrierter Schaltungen und Systeme, Dresden, Germany.
- [35] Böhme, H., Klingauf, W. and Telkamp, G., 2003: JControl – Rapid Prototyping und Design Reuse mit Java. In E.I.S.-Workshop - Entwurf Integrierter Schaltungen und Systeme, Erlangen, Germany.
- [36] Foundation for Intelligent Physical Agents (FIPA), 2002. SC00037J FIPA Communicative Act Library Specification. Geneva, Switzerland: Foundation for Intelligent Physical Agents.
- [37] Foundation for Intelligent Physical Agents (FIPA), 2002. SC00061G FIPA ACL Message Structure Specification. Geneva, Switzerland: Foundation for Intelligent Physical Agents.

**Dr. Kay Smarsly** was born in Siegen (Germany) in 1976. He graduated at Ruhr-University Bochum (Germany), where he received his academic engineer's degree ("Dipl.-Ing.") in 2002 and his PhD ("Dr.-Ing.") in civil and environmental engineering in 2008. In his major fields of research – structural health monitoring and artificial intelligence research – he recently received the "Zerna Prize", the "Built on IT Award", bestowed by the German Federal Ministry of Economics and Technology, and was awarded for his teaching activities with the "RUBel Award".

**Dr. Smarsly** is a reviewer for different academic journals and research foundations, and was an elected fellow of the prestigious Ruhr-University Research School (RURS). He is a member of various associations that are related to civil engineering and computer science, e.g. the International Association of Computer Science and Information Technology (IACSIT), the American Society of Civil Engineers (ASCE) and the European Group for Intelligent Computing in Engineering (EG-ICE).